



12/1  
UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/465,143	12/16/1999	RODGER W. TIGHE	P/3437-2	8524
2352	7590	03/18/2004	EXAMINER	
OSTROLENK FABER GERB & SOFFEN 1180 AVENUE OF THE AMERICAS NEW YORK, NY 100368403			NGUYEN, CHAU T	
		ART UNIT		PAPER NUMBER
		2176		61
DATE MAILED: 03/18/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	09/465,143	TIGHE, RODGER W.
	<b>Examiner</b>	<b>Art Unit</b>
	Chau Nguyen	2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 24 December 2003.

2a) This action is **FINAL**.                    2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-115 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1-115 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____.
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____.

## DETAILED ACTION

1. Amendment A, filed on 12/24/2003, has been entered..
2. Claims 1-115 are pending. Claims 1, 53, 59-65, 103, and 109-115 are independent claims.

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. With respect to the rejection of each dependent claim below, the preceding rejection(s) of the relevant base claim(s) is incorporated therein.

5. **Claims 1-15, 17-22, 24-29, 31, 38-44, 48, 50-51, 53-60, 62-81, 88, 90-95, 98, 100-101, 103-110, 112-115** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Number 6,314,415 B1 to Mukherjee, issued November 6, 2001, filed November 4, 1998 in view of Capsoft Inc., "HotDocs 5 Features", downloaded on April

13, 1999 from [www.capsoft.com/products/hotdocs/features.html](http://www.capsoft.com/products/hotdocs/features.html), provided in applicant's Information Disclosure Statement filed March 8, 2000 (hereinafter "HotDocs").

Regarding **independent claim 1**, Mukherjee teaches a user interface and controller comprising a programming environment having a plurality of programming functionalities. (Mukherjee, col. 5, line 29 – col. 6, line 4: "Display generator 123 operates under the control of application-specific shell 121, and may be implemented in C++, Visual Basic, JAVA, or another high level language."); Figures 3A-3L show a programming environment supporting certain types of forms that allow a browser to be used for the user interface)

Further, Mukherjee teaches the user interface and controller interfacing with a user by presenting a sequence of requests to a user so that the user provides information necessary to prepare the document. (Mukherjee, col. 5, lines 35-40: "In summary, display generator 123 generates graphical user interface components on computer display 130 in response to form definitions retrieved through SQL access component 124 and the firing of expert system rules executed by inference engine 125, which are in turn driven by user inputs received through computer display 130."); see also Mukherjee, col. 2, lines 45-51.)

Further, Mukherjee teaches a database supplying first and second data to the controller wherein the first data provides a format for requests to the user (Mukherjee,

Art Unit: 2176

col. 5, lines 35-39: form definitions) and the second data is for preparing text in a document (Mukherjee, col. 5, lines 41-44: "As a user enters information through computer display 130, various rules in the database will be "fired," thus changing the appearance of the display for future prompts").

Further, Mukherjee does not teach the user interface and controller in communication with a word processing application to control the word processing application to prepare and generate the document. However, HotDocs teaches in lines 22-25 of page 2 controlling a word processing application such as Microsoft Word to prepare and generate a document. Moreover, one of ordinary skill in the art would have recognized that Microsoft Word was a program which many users had on their computers, were comfortable using, and also that the format of a Microsoft Word document would be a format that could easily be shared and disseminated. Therefore, it would have been obvious to one of ordinary skill in the art to have the user interface and controller in communication with a word processing application to control the word processing application to prepare and generate the document.

Further, Mukherjee teaches the user interface and controller using combinations of the programming functionalities to prepare and generate the document (Mukherjee, col. 5, line 46 – col. 6, line 18: after a user has entered information in response to the dynamically changing prompts, forms could be electronically generated and stored for electronic filing).

Further, Mukherjee teaches the user interface and controller comprising a program for providing the sequence of requests to the user to obtain information from the user for preparation of the document. (Mukherjee, Figs. 3A – 3L.)

Further, Mukherjee teaches the user interface and controller comprising a program for further manipulating the document in response to information received from the user (Mukherjee, col. 5, lines 40-45; col. 6, lines 5-18), although Mukherjee does not teach using a word processing program for the manipulation. However, it would have been obvious to one of ordinary skill in the art to use a word processing program in view of HotDocs for the reasons noted above.

Further, Mukherjee teaches the second data from the database to draft the contents of the document (col. 6, lines 5-18, col. 8, lines 53-65, and col. 17, line 50 – col. 18, line 7: when all required questions have been answered, the user can save his or her answers in output forms populated with information entered by user):

Further, Mukherjee teaches that the sequence of requests is dynamically altered by the program in response to the information provided in response to previous requests, the program determining the next request of the dynamically alterable sequence of requests to be provided to the user and when and how to manipulate the document such that each request is the logical extension of information provided by the

user in response to previous requests and each request that is not the logical extension of information provided by the user is omitted. (Mukherjee, col. 5, lines 46-56: "As one example, if a user were to select a choice on display 130 indicating that his marital status is "single," one rule could be fired to assert a condition that the person is single, and another rule could be fired (based on that assertion) removing any fields from display 130 relating to spouses. As another example, if a user were to indicate in response to a query that she has three children, a rule could fire that generates three separate blocks of data fields on display 130, one for each child (i.e., name, social security number and date of birth for each child).")

Regarding **independent claim 53**, the rejection of claim 1 above is fully incorporated herein.

Further, Mukherjee does not teach the program making all changes to the document necessitated by the information solicited by the sequence of requests. However, in view of HotDocs' teaching of a reusable document template (page 1, 4<sup>th</sup> line from the bottom to page 2, line 1), it would have been obvious to one of ordinary skill in the art to implement the program making all changes to the document necessitated by the information solicited by the sequence of requests because one of ordinary skill in the art would have recognized that a reusable template provided the benefit of efficiency and that in order to produce the document required by the user it

would have been necessary to make all changes to the document necessitated by the information solicited by the sequence of requests.

Further, Mukherjee teaches inserting text into the document at all location in the document affected by the information inasmuch as this limitation would have been obvious to one of ordinary skill in the art from Mukherjee's teaching of generating the document based on the information (Mukherjee, col. 6, lines 5-8) because one of ordinary skill in the art would have recognized that the user would have wanted all responses to requests for information reflected in the document.

Further, Mukherjee does not explicitly teach that text is different in at least one location from the other locations, but one of ordinary skill in the art would have recognized from the fact that Mukherjee teaches entering different sets of information (Mukherjee, Figs. 3A – 3L), it would have been desirable to have different text reflecting the different information at different locations. Therefore, it would have been obvious to one of ordinary skill in the art to implement a system wherein text is different in at least one location from the other locations.

Regarding **independent claim 59**, the rejection of claim 1 above is fully incorporated herein.

Art Unit: 2176

Further, Mukherjee does not teach providing a confirmation dialog to allow the user to confirm responses to selected requests. However, notice is taken that input confirmation dialog boxes were well known in the art prior to applicant's claimed invention, and it was further well known that such dialog boxes provided the advantage of ensuring that a user provided the correct input. Moreover, one of ordinary skill in the art would have recognized that providing such a box would have ensured that the document was generated with the correct content. Therefore, it would have been obvious to one of ordinary skill in the art to implement the program providing a confirmation dialog to allow the user to confirm responses to selected requests.

Further, Mukherjee does not teach making a change to the document after confirming the response was correct, or resoliciting information if the response was incorrect. However, one of ordinary skill in the art would have recognized that if the response was correct, the user would have wanted the document changed accordingly, and if the response was incorrect, the user would have wanted the opportunity to correct it. Therefore, it would have been obvious to one of ordinary skill in the art to implement the program making a change to the document after confirming the response was correct, or resoliciting information if the response was incorrect.

Regarding **independent claim 60**, the rejection of claim 1 above is fully incorporated herein.

Art Unit: 2176

Further, Mukherjee does not teach that the program makes at least one change necessitated by responses to the sequence of requests is being presented to the user. However, in view of the obviousness of using Microsoft Word, discussed above, it further would have been obvious to one of ordinary skill in the art to make changes to the document in response to the sequence of requests because one of ordinary skill in the art would have recognized first that Word enabled changes to be made as soon as information was received via macro functionality and secondly that this would have enabled the user to be working with the most accurate document possible.

Further, Mukherjee teaches a template containing document provisions, inasmuch as the form documents taught by Mukherjee are equivalent to templates and would have contained unaltered text as well as locations into which text was inserted. (Mukherjee, col. 6, lines 8-18.) Mukherjee does not teach the template included in a word processing application, but this limitation would have been obvious to one of ordinary skill in the art in view of HotDocs for the reasons discussed above.

Further, Mukherjee does not teach that the user can modify the template. However, HotDocs teaches document templates (4<sup>th</sup> line from the bottom of page 1 to the first line of page 2) that could be edited. Moreover, one of ordinary skill in the art would have recognized that the user might wish to make changes to a template that was only partially suited to the document the user wished to produce, e.g., the template might not contain all the provisions the user required. Therefore, it would have been

Art Unit: 2176

obvious to one of ordinary skill in the art that the user can modify the template and add provisions to it.

Regarding **independent claim 62**, the rejection of claim 60 above is fully incorporated herein.

Further, Mukherjee does not teach the database containing at least one textual provision for insertion into the document and allowing the user to modify the textual provision and replace it in the database with the modified version. However, HotDocs teaches storing document templates, which inherently would have contained textual provisions in file systems, equivalent to databases (page 1, 4<sup>th</sup> line from the bottom to page 2 line 1) and also teach that these templates could be modified and saved. One of ordinary skill in the art would have recognized that being able to reuse textual provisions, and being able to modify and save them without creating a new one from scratch, would have provided advantageous efficiency to the user. Therefore, it would have been obvious to one of ordinary skill in the art to implement the database containing at least one textual provision for insertion into the document and allowing the user to modify the textual provision and replace it in the database with the modified version.

Regarding **independent claim 63**, the rejection of claim 60 above is fully incorporated herein.

Further, Mukherjee teaches the sequence of requests being presented on a form displayed on a display device. (Mukherjee, Figs. 3A-3L.)

Further, Mukherjee discloses a form having at least one entry location for inserting a number. (Mukherjee, Fig. 3H.)

Further, Mukherjee does not teach a plurality of fields of increasing order so that the user can insert a number into a selected field of highest selected order without inserting a number into lower order fields and wherein the lower order fields are automatically populated with a zero in the absence of user insertion of a number into such lower order fields. However, one of ordinary skill in the art would have recognized that tax documents, insurance documents, and retirement plan documents taught by Mukherjee (Mukherjee, col. 1, lines 22-30) would have been likely to have a plurality of fields presenting a range of numbers (for example, an insurance document might have listed the deductibles at various claim levels). One of ordinary skill in the art would have also recognized that if a user inserted a number at a selected order but did not insert a number in lower orders that the value of the lower orders was likely to be zero. (For example, inserting a value of a deductible at a certain claim level but not at others would have implied that there was no deductible at those claim levels.) Further, one of ordinary skill in the art would have recognized that automatically filling in fields with zeros would have saved the user the trouble of doing so. Therefore, it would have been obvious to one of ordinary skill in the art to implement the recited claim limitation.

Regarding **independent claim 64**, the rejection of claim 60 above is fully incorporated herein.

Further, Mukherjee teaches a user control comprising at least one form allowing a user to input a date, the form comprising at least one entry are for month, day, and year. (Mukherjee, Fig. 3K.)

Regarding **independent claim 65**, the rejection of claim 1 above is fully incorporated herein.

Further, Mukherjee teaches using a programming environment (Mukherjee, col. 5, lines 29-30) although Mukherjee does not teach the programming environment controlling the word processing application. However, HotDocs teaches in lines 22-25 of page 2 controlling a word processing application such as Microsoft Word to prepare and generate a document. Moreover, one of ordinary skill in the art would have recognized that Microsoft Word was a program which many users had on their computers, were comfortable using, and also that the format of a Microsoft Word document would be a format that could easily be shared and disseminated. Therefore, it would have been obvious to one of ordinary skill in the art to have the programming environment controlling the word processing application.

Regarding **independent claim 103**, the rejection of claims 53 and 65 above are both fully incorporated herein.

Regarding **independent claim 109**, the rejection of claims 59 and 65 above are both fully incorporated herein.

Regarding **independent claim 110**, the rejection of claims 60 and 65 above are both fully incorporated herein.

Regarding **independent claim 112**, the rejection of claims 62 and 65 above are both fully incorporated herein.

Regarding **independent claim 113**, the rejection of claims 63 and 65 above are both fully incorporated herein.

Regarding **independent claim 114**, the rejection of claims 64 and 65 above are both fully incorporated herein.

Regarding **independent claim 115**, the rejection of claim 1 above is fully incorporated herein.

Regarding **dependent claims 2 and 66**, Mukherjee teaches the user interface and controller directly manipulating the document to prepare and generate it. (Mukherjee, col. 5, lines 29-31; col. 6, lines 5-8.)

Regarding **dependent claims 3 and 67**, Mukherjee teaches a template containing text that is unaltered as well as at least one location in the template defining a location in the document for the insertion of text into the document, inasmuch as the form documents taught by Mukherjee are equivalent to templates and would have contained unaltered text as well as locations into which text was inserted. (Mukherjee, col. 6, lines 8-18.) Mukherjee does not teach the template included in a word processing application or the word processing application inserting text into the document, but this limitation would have been obvious to one of ordinary skill in the art in view of HotDocs for the reasons discussed above regarding claim 1.

Regarding **dependent claims 4 and 68**, Mukherjee teaches the database providing the second data which is inserted directly in the document. (Mukherjee, col. 6, lines 5-14). Further, Mukherjee does not teach, but it would have been obvious to one of ordinary skill in the art to instruct the word processing application to insert the text into the document for the reasons discussed above regarding claim 1.

Regarding **dependent claims 5 and 69**, Mukherjee does not teach the word processing application inserting the second data into the document, although as noted above regarding claim 5, Mukherjee does teach the insertion of second data into the document. Moreover, for the reasons discussed above regarding claim 1, it would have been obvious to one of ordinary skill in the art to implement the word processing application inserting the second data into the document.

Regarding **dependent claims 6 and 70**, Mukherjee teaches providing data directly into the document in response to commands from the user interface and controller. (Mukherjee, col. 6, lines 8-14: "Based on form definition information in database 129, all the necessary forms (e.g., employment application forms, tax withholding forms, health benefit forms) can be generated that match a particular company's format requirements (and the government's format and content requirements) even though the user did not separately create data for each form.")

Regarding **dependent claims 7 and 71**, Mukherjee teaches the sequence of requests presented to the user by screen displays comprising a plurality of forms presenting the requests to the user. (Mukherjee, Figs 3A – 3L.)

Regarding **dependent claims 8 and 72**, Mukherjee teaches the first data comprising data used in determining the form of the plurality of forms displayed and the content of each form displayed to solicit information from the user to prepare the document. (Mukherjee, col. 5, lines 35-40.)

Regarding **independent claims 9 and 73**, Mukherjee teaches that the forms are dynamically alterable in response to information provided in response to previous requests. (Mukherjee, col. 8, lines 53-65.)

Regarding **dependent claims 10 and 74**, Mukherjee teaches the forms comprising a group of screen displays each having a specified format (Mukherjee, Figs. 3A – 3L), the format of at least one form comprising at least one dynamically alterable request and an entry area for responding to the at least one request. (Mukherjee, Fig 3B; col. 8, lines 39-53.)

Regarding **dependent claims 11 and 77**, Mukherjee teaches the program executing at least one logical function inasmuch as the functionality described above with reference to claim 10 would have required the execution of at least one logical function.

Regarding **dependent claim 12**, Mukherjee teaches the program comprising a program logic inasmuch as a program written in the languages disclosed by Mukherjee (Mukherjee, col. 5, lines 29-31) would have comprised a program logic and moreover a program logic would have been necessary for the program to execute and perform the functions described by Mukherjee. Further, Mukherjee teaches an insert test function that inserted text into the document upon command when the program logic had received all information relating to a specified provision of the document. (Mukherjee, col. 6, lines 5-14.)

Regarding **dependent claim 13**, Mukherjee teaches the program logic executing at least one logical function inasmuch as Mukherjee teaches displaying different

interface components depending on the firing of rules, which would have required the execution of at least one logical function. (Mukherjee, col. 5, lines 41-44.)

Further, Mukherjee teaches the insert text function executing at least one logical function inasmuch as the different forms taught by Mukherjee (Mukherjee, col. 6, lines 9-11) would have required the insert text function to determine what text could or should appropriately be inserted into the form.

Regarding **dependent claim 14**, Mukherjee teaches the program logic executing at least one logical function based on information obtained from the user in response to the forms inasmuch as Mukherjee teaches the program deciding which screens to display based on user input. (Mukherjee, col. 5, lines 41-43.)

Further, Mukherjee inherently would have had an insert text function executing at least one logical function based on information obtained from the user in response to the forms inasmuch as Mukherjee teaches choosing different kinds of forms and inputting information in the right location on the different forms which inherently would have required executing at least one logical function based on information obtained from the user in response to the forms. (Mukherjee, col. 6, lines 8-18.)

Regarding **dependent claim 15**, Mukherjee teaches the program executing a plurality of logical functions or combinations of logical functions inasmuch as Mukherjee

Art Unit: 2176

teaches the program executing the progression of screens that would have required execution of logical functions in Figs. 3A – 3L.

Regarding **dependent claims 17 and 76**, (note that the rejection of claim 75, from which claim 76 depends, is provided below) the insert text function modifying the document at all locations affected by the information relating to a specified issue in the document. However, one of ordinary skill in the art would have recognized that the user would have wanted the proper text relating to an issue inserted at all points in the document where the issue arose. Therefore, it would have been obvious to one of ordinary skill in the art to have the insert text function modifying the document at all locations affected by the information relating to a specified issue in the document.

Regarding **dependent claim 18**, Mukherjee does not explicitly teach the insert text function returning control to the program logic once text has been inserted into the document. However, one of ordinary skill in the art would have recognized that once text had been inserted into the document it would have been necessary to return program control so that the user could display, print, distribute, or make use of in some other way, the document. Also, one of ordinary skill in the art would have known that it is good programming practice for a function to return program control once the function is executed. Therefore, it would have been obvious to one of ordinary skill in the art to have the insert text function returning control to the program logic once text has been inserted into the document.

Regarding **dependent claim 19**, Mukherjee teaches that the forms communicate with the database to obtain information necessary for preparation of the document and the database provides the first data to the forms necessary to prepare a respective form to solicit information from the user for the preparation of the document. (Mukherjee, col. 5, lines 34-40: "In summary, display generator 123 generates graphical user interface components on computer display 130 in response to form definitions retrieved through SQL access component 124 and the firing of expert system rules executed by inference engine 125, which are in turn driven by user inputs received through computer display 130.")

Regarding **dependent claim 20**, Mukherjee does not teach the insert text function manipulating the document by providing instructions to the word processing application to manipulate the document. However, as noted above regarding claim 1, it would have been obvious in view of HotDocs to have the program provide instructions to a word processing program to prepare and generate the document, and inasmuch as one of ordinary skill in the art would have recognized that document preparation and generation included inserting text, it further would have been obvious to one of ordinary skill in the art to implement the insert text function manipulating the document by providing instructions to the word processing application to manipulate the document.

Regarding **dependent claim 21**, Mukherjee teaches the insert text function directly manipulating the document. (Mukherjee, col. 6, lines 5-14.)

Regarding **dependent claim 22**, Mukherjee teaches obtaining second data from the database for displaying in a respective form at least one textual provision for insertion into the document. (Mukherjee, col. 8, lines 39-43: "FIG. 3B shows the screen of FIG. 3A after a user has selected a filing type (appointment), residency status (resident) and a multi-producer jurisdiction (Pennsylvania), thus enabling the "number of producers" prompt 304 and storing a default value of 1 in the prompt.")

Regarding **dependent claim 24**, Mukherjee teaches that the user interface and controller comprises a programmed computer executing an application created using Visual Basic. (Mukherjee, col. 5, lines 29-31.)

Regarding **dependent claim 25**, Mukherjee does not teach that the word processing application comprises Microsoft Word, but this claim limitation would have been obvious to one of ordinary skill in the art in view of HotDocs as discussed above regarding claim 1.

Regarding **dependent claims 26 and 79**, Mukherjee does not teach the program including at least one confirmation dialog to interface with the user to confirm at least one response to a previous request. However, notice is taken that input confirmation

Art Unit: 2176

dialog boxes were well known in the art prior to applicant's claimed invention, and it was further well known that such dialog boxes provided the advantage of ensuring that a user provided the correct input. Moreover, one of ordinary skill in the art would have recognized that providing such a box would have ensured that the document was generated with the correct content. Therefore, it would have been obvious to one of ordinary skill in the art to implement the program including at least one confirmation dialog to interface with the user to confirm at least one response to a previous request.

Regarding **dependent claims 27 and 78**, Mukherjee does not teach that after a user confirmed that a response to the confirmation dialog was correct, the document is manipulated to alter the document in response to the information that is the subject of the confirmation dialog. However, one of ordinary skill in the art would have recognized that if the response was correct, the user would have wanted the document manipulated accordingly. Therefore, it would have been obvious to one of ordinary skill in the art that after a user confirmed that a response to the confirmation dialog was correct, the document is manipulated to alter the document in response to the information that is the subject of the confirmation dialog.

Regarding **dependent claims 28 and 80**, Mukherjee does not teach that the user can manipulate the template and make changes to the template. However, HotDocs teaches document templates (4<sup>th</sup> line from the bottom of page 1 to the first line of page 2) that could be edited. Moreover, one of ordinary skill in the art would have

Art Unit: 2176

recognized that the user might wish to make changes to a template that was only partially suited to the document the user wished to produce. Therefore, it would have been obvious to one of ordinary skill in the art that the user can manipulate the template and make changes to the template.

Regarding **dependent claims 29 and 81**, Mukherjee does not teach that the changes are optionally permanent. However, notice is taken that the ability to display and print a document in applications such as Microsoft Word and then decide whether to save the changes was well known in the art prior to applicant's claimed invention. Moreover, one of ordinary skill in the art would have recognized that this functionality gave the user the ability to make changes that would only be used once without affecting the permanent template. Therefore, it would have been obvious to one of ordinary skill in the art to make the changes are optionally permanent.

Regarding **dependent claims 31 and 88**, Mukherjee teaches that the program includes text for insertion into the document. (Mukherjee, col. 6, lines 9-11.)

Regarding **dependent claim 38**, Mukherjee does not explicitly teach at least one form having an entry location for inserting a monetary amount, but one of ordinary skill in the art would have recognized that monetary fields would have been necessary for the insurance, retirement plan, and tax documents taught by Mukherjee (Mukherjee, col.

1, lines 22-30), and therefore, it would have been obvious to one of ordinary skill in the art at least one form having an entry location for inserting a monetary amount.

Further, Mukherjee does not teach a plurality of fields of increasing order so that the user can insert a number into a selected field of highest selected order without inserting a number into lower order fields and wherein the lower order fields are automatically populated with a zero in the absence of user insertion of a number into such lower order fields. However, one of ordinary skill in the art would have recognized that tax documents, insurance documents, and retirement plan documents would have been likely to have a plurality of fields presenting a range of numbers (for example, an insurance document might have listed the deductibles at various claim levels). One of ordinary skill in the art would have also recognized that if a user inserted a number at a selected order but did not insert a number in lower orders that the value of the lower orders was likely to be zero. (For example, inserting a value of a deductible at a certain claim level but not at others would have implied that there was no deductible at those claim levels.) Further, one of ordinary skill in the art would have recognized that automatically filling in fields with zeros would have saved the user the trouble of doing so. Therefore, it would have been obvious to one of ordinary skill in the art to implement the recited claim limitation.

Regarding **dependent claims 39 and 91**, Mukherjee teaches that the forms comprise at least one form allowing a user to input a date, the form comprising at least one entry are for month, day, and year. (Mukherjee, Fig. 3K.)

Regarding **dependent claim 40**, Mukherjee teaches Microsoft Access as the database. (Mukherjee, col. 4, lines 60-61.)

Regarding **dependent claim 41**, Mukherjee teaches a digital computer comprising all of the elements recited in claim 41 except for a word processor (Mukherjee, Fig. 1), and a word processor would have been obvious to one of ordinary skill in the art in view of HotDocs for the reasons stated above regarding claim 1.

Regarding **dependent claims 42 and 92**, Mukherjee does not teach that the word processing application embeds in the document code to enable the user to continue modifying the document using only the word processing application. However, as noted above regarding claim 1 would have been obvious to one of ordinary skill in the art in view of HotDocs to use Microsoft Word, and inherent in the use of Microsoft Word would have been embedding in the document code to enable the user to continue modifying the document using only the word processing application inasmuch as would have been obvious to one of ordinary skill in the art in view of HotDocs to create a Microsoft Word document that inherently would have had such embedded codes.

Regarding **dependent claims 43 and 93**, Mukherjee does not teach that the program allows the user to make changes to the document while the sequence of requests is being presented to the user. However, in view of the obviousness of using Microsoft Word, discussed above regarding claim 1, it further would have been obvious to one of ordinary skill in the art to allow the user to make changes directly to the document while the sequence of event was presented to the user because one of ordinary skill in the art would have recognized first that Word enabled such functionality and secondly that this functionality would have allowed the user to make changes that might not have been enabled by the sequence of events.

Regarding **dependent claims 44 and 94**, Mukherjee teaches that the sequence of events is dynamically alterable such that the user is not provided any redundant requests for information nor any requests inconsistent with responses to previous requests. (Mukherjee, col. 5, lines 43:45: "The dynamically changing nature of the graphical user interface avoids prompting the user for redundant or irrelevant information.")

Regarding **dependent claims 48 and 98**, Mukherjee does not teach the program determining and directing the word processing application as to where in the document textual provisions are to be inserted, locations for the insertion of textual provisions in the document being dynamically added or changed by the program directing the word processing application. However, as noted above regarding claim 1, it would have been

obvious to one of ordinary skill in the art over Mukherjee in view of HotDocs to have the program direct a word processing application as to where in the document textual provisions are to be inserted. Moreover, inasmuch as Mukherjee teaches dynamically changing the information to be placed in the document as noted above regarding claim 1, one of ordinary skill in the art would have recognized the need to have the locations for the insertion of textual provisions in the document be dynamically added or changed. Therefore, it would have been obvious to one of ordinary skill in the art to implement this claim limitation.

Regarding **dependent claims 50 and 100**, Mukherjee teaches that the program includes at least one mathematical calculation routine inasmuch as such a routine would have been necessary to support the "Question Count" field depicted in Fig. 3F.

Regarding **dependent claims 51 and 101**, Mukherjee teaches the database optionally storing a table of responses to the requests to prepare a document wherein the table of responses can be displayed to a user so that the user can change responses whereby the document is redrafted based on the responses. (Mukherjee, col. 18, lines 20-57: "As one example, a first table can be used to store a complete list of transactions (e.g., hire an employee, fire an employee, etc.), each having a transaction code and description . . . . Assuming the information described above is stored in separate tables, it is of course possible to convert the data into third normal form and store it in a relational database for easier access and optimization.

Art Unit: 2176

Additionally, although the expert system rules can be stored in the database, it may be preferable to extract the rules, combine them with other general rules that are application-independent (e.g., for setting up the display and initializing data values), and compile them into a binary file that can be executed by a PROLOG inference engine.”)

Regarding **dependent claims 54-56 and 104-106**, Mukherjee teaches the sequence of requests comprising a plurality of screen displays (Mukherjee, Figs. 3A – 3L), with subsequent screen displays being dynamically alterable in response to information obtained from a previous screen display. (Mukherjee, Fig 3B; col. 8, lines 39-53.)

Regarding **dependent claims 57 and 107**, Mukherjee teaches that a response to at least one of the sequence of requests does not result in alteration of the document until a response is provided to at least one subsequent request inasmuch as Mukherjee teaches that the document is not altered at all until all requests have been responses to. (Mukherjee, col. 6, lines 5-8.)

Regarding **dependent claims 58 and 108**, Mukherjee teaches that the program dynamically alters a screen display in response to information obtained in response to a request (Mukherjee, Fig 3B; col. 8, lines 39-53) and the document is altered to reflect the information obtained. (Mukherjee, col. 6, lines 5-8.)

Regarding **dependent claim 75**, Mukherjee does not teach inserting text into the document when all information relating to a specified provision has been received. However, in view of the obviousness of using Microsoft Word, discussed above, it further would have been obvious to one of ordinary skill in the art to inserting text into the document when all information relating to a specified provision has been received because one of ordinary skill in the art would have recognized first that Word enabled changes to be made as soon as information was received via macro functionality and secondly that this would have enabled the user to be working with the most accurate document possible.

Regarding **dependent claim 90**, Mukherjee does not teach a plurality of fields of increasing order so that the user can insert a number into a selected field of highest selected order without inserting a number into lower order fields and wherein the lower order fields are automatically populated with a zero in the absence of user insertion of a number into such lower order fields. However, one of ordinary skill in the art would have recognized that tax documents, insurance documents, and retirement plan documents taught by Mukherjee (Mukherjee, col. 1, lines 22-30) would have been likely to have a plurality of fields presenting a range of numbers (for example, an insurance document might have listed the deductibles at various claim levels). One of ordinary skill in the art would have also recognized that if a user inserted a number at a selected order but did not insert a number in lower orders that the value of the lower orders was likely to be zero. (For example, inserting a value of a deductible at a certain claim level

Art Unit: 2176

but not at others would have implied that there was no deductible at those claim levels.) Further, one of ordinary skill in the art would have recognized that automatically filling in fields with zeros would have saved the user the trouble of doing so. Therefore, it would have been obvious to one of ordinary skill in the art to implement the recited claim limitation.

Regarding **dependent claim 95**, Mukherjee does not teach manipulating the document while requests are being presented to the user. However, one of ordinary skill in the art would have recognized that the user would have wanted to see the document updated as much as possible to take into account the most recent changes so that the user would not be working with and viewing an out-of-date document. Therefore, it would have been obvious to one of ordinary skill in the art to implement manipulating the document while requests are being presented to the user.

6. **Claim 16** is rejected under 35 U.S.C. 103(a) as being unpatentable over Mukherjee in view of HotDocs and further in view of Microsoft Corporation, *Microsoft Visual Basic Programmer's Guide* (1995), pages 137-38.

Mukherjee does not teach that the plurality of logical functions include the SELECT CASE statement of Visual Basic, although Mukherjee does teach program logic including logical functions as set forth above regarding claim 11, and also teaches program logic implemented in Visual Basic. (Mukherjee, col. 5, lines 29-31.) *Microsoft Visual Basic Programmer's Guide* teaches on page 137 that the SELECT CASE

statement is a more efficient, readable programming construct than its alternative, the IF...THEN...ELSE statement. Therefore, it would have been obvious to one of ordinary skill in the art to extend Mukherjee's teaching of the use of Visual Basic to implement logical functions to implement the plurality of logical functions include the SELECT CASE statement of Visual Basic.

7. **Claim 23** is rejected under 35 U.S.C. 103(a) as being unpatentable over Mukherjee in view of HotDocs and IQ Doc, LLC, *Intelligent Document Assembly* (April 14, 1999), provided in applicant's Information Disclosure Statement filed March 8, 2000.

Mukherjee does not teach that the document comprises a finance commitment letter. However, on page 1 of [www.iqdocs.com/dcument.htm](http://www.iqdocs.com/dcument.htm) [sic], in the section entitled "IQ Docs 'The Smart Document Assembler'", *Intelligent Document Assembly* teaches an automatic document assembly system for commercial loan documents. Moreover, one of ordinary skill in the art would have understood from *Intelligent Document Assembly* that it taught the benefit of relieving users of the need to repeatedly create language that was used over and over for commercial lending documents. Moreover, one of ordinary skill in the art would have recognized that a finance commitment letter was such a document. Therefore, it would have been obvious to one of ordinary skill in the art to have the document comprise a finance commitment letter.

Art Unit: 2176

8. **Claims 30, 32-34, 52, 82-85, and 102** are rejected under 35 U.S.C. 103(a) as being unpatentable over Mukherjee in view of HotDocs and further in view of U.S. Patent Number 6,473,892 B1 to Porter, issued October 29, 2002, filed December 31, 1998.

Regarding **dependent claims 30 and 82**, Mukherjee does not teach the program allowing the user to generate at least one alternative provision for insertion into the document. However, in the context of an automated document assembly system, Porter teaches generating alternative provisions and also teaches that this step provides the benefit of accommodating special cases that can arise in a legal document. (Porter, col. 11, lines 4-14: "For example, an IF/ELSE statement 124 within the block of code 112 defining the text of first paragraph 114 selects between two versions of a phrase depending on whether the loan is made in the state of Oklahoma (OK). The decision is based on the value of the DocST property of the input data object 20 (FIG. 2) indicating the state having jurisdiction over the transaction. The state of Oklahoma has a unique legal definition for the term "Loan Fund" and that terminology should be used in loan documents over which Oklahoma has jurisdiction.") Therefore, it would have been obvious to one of ordinary skill in the art to implement the program allowing the user to generate at least one alternative provision for insertion into the document.

Regarding **dependent claims 32-34 and 83-85**, Mukherjee does not teach that the user has the option to use the at least one alternative provision once or save it

permanently for future use, or that a single use provision is inserted into the document and not saved or that a saved provision for future use is added to the database. However, notice is taken that the ability to display and print a document in applications such as Microsoft Word and then decide whether to save the changes was well known in the art prior to applicant's claimed invention. Moreover, one of ordinary skill in the art would have recognized that this functionality gave the user the ability to make changes that would only be used once without affecting the permanent document. Also, one of ordinary skill in the art would have recognized that a single use provision by definition should have been inserted into the document and not saved and that a saved provision for future use by definition should be added to the database. Therefore, it would have been obvious to one of ordinary skill in the art to give the user the option to use the at least one alternative provision once or save it permanently for future use, and that a single use provision is inserted into the document and not saved or that a saved provision for future use is added to the database.

Regarding **dependent claims 52 and 102**, Mukherjee does not teach storing commentary together with selected alternate provisions, but Mukherjee does teach storing help strings, equivalent to commentary, for certain forms. (Mukherjee, col. 18, lines 36-38.) In view of the obviousness of alternative provisions discussed above regarding claim 30, one of ordinary skill in the art would have recognized that help, or commentary, would have helped the user understand different alternative provisions,

and therefore, it would have been obvious to one of ordinary skill in the art to extend Mukherjee's teaching of help, or commentary, to alternative provisions.

9. **Claims 35-37, 47, 61, 86-87, 89, 97 and 111** are rejected under 35 U.S.C. 103(a) as being unpatentable over Mukherjee in view of HotDocs, Porter and U.S. Patent Number 5,729,751 to Schoolcraft, issued March 17, 1998.

Regarding **independent claim 61**, the rejection of claim 1 above is fully incorporated herein.

Further, Mukherjee does not teach that the program makes at least one change necessitated by responses to the sequence of requests is being presented to the user. However, in view of the obviousness of using Microsoft Word, discussed above, it further would have been obvious to one of ordinary skill in the art to make changes to the document in response to the sequence of requests because one of ordinary skill in the art would have recognized first that Word enabled changes to be made as soon as information was received via macro functionality and secondly that this would have enabled the user to be working with the most accurate document possible.

Further, Mukherjee does not teach that the database has stored therein textual provisions and adding alternative textual provisions to a document and giving the user the option of storing them in a database. However, Schoolcraft teaches inserting textual provisions stored in a database as well as adding permanent alternative textual

Art Unit: 2176

provisions (Schoolcraft, col. 9, lines 29-42), and Porter would have provided motivation to one of ordinary skill in the art as noted above regarding claim 30. Therefore, it would have been obvious to one of ordinary skill in the art that the database has stored therein textual provisions and adding alternative textual provisions to a document and giving the user the option of storing them in a database.

Regarding **independent claim 111**, the rejection of claims 61 and 65 above are both fully incorporated herein.

Regarding **dependent claim 35**, Mukherjee does not teach that the database has stored therein alternative provisions for insertion in the document. However, Schoolcraft teaches inserting alternative provisions stored in a database (Schoolcraft, col. 9, lines 29-42), and Porter would have provided motivation to one of ordinary skill in the art as noted above regarding claim 30. Therefore, it would have been obvious to one of ordinary skill in the art that the database has stored therein alternative provisions for insertion in the document.

Regarding **dependent claims 36 and 86**, Mukherjee does not teach that alternative textual provisions in the database area accessible by the user for modifying the alternative textual provisions or adding a further alternative textual provision. However, Schoolcraft teaches that a user can access a database to add alternative provisions. (Schoolcraft, col. 9, lines 37-42.) Moreover, one of ordinary skill in the art

Art Unit: 2176

would have recognized that the user might have wished to modify alternative provisions as the law or other standard terms changed, just as a user may have wished to add alternative provisions to accommodate new situations. Therefore, it would have been obvious to one of ordinary skill in the art to implement alternative textual provisions in the database area accessible by the user for modifying the alternative textual provisions or adding a further alternative textual provision.

Regarding **dependent claim 37 and 89**, Mukherjee does not teach accessing directly the alternative textual provisions in the database when not responding to the sequence of requests. However, Schoolcraft does teach such access as noted above regarding claim 36. Moreover, for the reasons noted above regarding claim 36, one of ordinary skill in the art would have been motivated to provide such access. Therefore, it would have been obvious to one of ordinary skill in the art to access directly the alternative textual provisions in the database when not responding to the sequence of requests.

Regarding **dependent claims 47 and 97**, Mukherjee does not teach that the database comprises a plurality of alternative provisions for the document wherein the alternative provisions are evaluated in response to user-provided information to determine which of the provisions to display in a form. However, Schoolcraft teaches storing a plurality of alternative provisions stored in a database (Schoolcraft, col. 9, lines 29-42), and Porter would have provided motivation to one of ordinary skill in the art as

Art Unit: 2176

noted above regarding claim 30 to evaluate the alternative provisions and display one in a form. Therefore, it would have been obvious to one of ordinary skill in the art to implement the limitations recited herein.

Regarding **dependent claim 87**, the rejection of claim 83 above is fully incorporated herein.

10. **Claims 45-46, 49, 96, and 99** are rejected under 35 U.S.C. 103(a) as being unpatentable over Mukherjee in view of HotDocs and further in view of U.S. Patent Number 5,367,619 to Dipaolo et al., issued November 22, 1994.

Regarding **dependent claim 45**, Mukherjee does not teach that the system manipulates the document as requests are presented to the user. However, Dipaolo et al. teach propagating changes necessitated by a change in one form field to fields throughout a document (Dipaolo et al., col. 6, lines 39-51). One of ordinary skill in the art would have recognized that this teaching provided the advantage of allowing the user to work with a document that had correct values throughout. Therefore, it would have been obvious to one of ordinary skill in the art to have the system manipulate the document as requests are presented to the user.

Regarding **dependent claims 46 and 96**, Mukherjee does not teach providing a simultaneous display to the user showing the requests and at least a portion of the

Art Unit: 2176

document as it is being prepared. However, Dipaolo et al. teach showing both menu menus for selecting form values and a list of permissible values to be selected side by side. (Dipaolo et al., col. 6, lines 16-24.) Moreover, one of ordinary skill in the art would have recognized the advantage of allowing the user to see the actual field for which a selection was being made as well as other impacted fields at the same time the user was entering values for the field. Therefore, it would have been obvious to one of ordinary skill in the art to provide a simultaneous display to the user showing the requests and at least a portion of the document as it is being prepared.

Regarding **dependent claims 49 and 99**, Mukherjee does not teach that the document is dynamically altered in response to information provided by the user in response to the sequence of requests to modify the document to present the information provided by the user. However, Dipaolo et al. teach dynamically propagating changes necessitated by a change in one form field to fields throughout a document (Dipaolo et al., col. 6, lines 39-51). One of ordinary skill in the art would have recognized that this teaching provided the advantage of allowing the user to work with a document that had correct values throughout. Therefore, it would have been obvious to one of ordinary skill in the art that the document is dynamically altered in response to information provided by the user in response to the sequence of requests to modify the document to present the information provided by the user.

### ***Conclusion***

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

<b>U.S. Patent Number</b>	<b>Name</b>	<b>Issue Date</b>	<b>File Date</b>	
6,345,278 B1	Hitchcock et al.	2/5/02	6/3/99	
6,226,656 B1	Zawadzki et al.	5/1/01	11/12/98	
6,134,568	Tonkin	10/17/00	10/30/98	
5,893,914	Clapp	4/13/99	3/17/97	
5,732,221	Feldon et al.	3/24/98	n/a	

### **Response to Arguments**

In the remarks, Applicant argued in substance that

(A) HotDocs does not take full advantage of a programming environment to control a word processing environment and a data base to manipulate a document to actually draft the contents of the document.

As to point (A), in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, Mukherjee teaches a user interface and controller comprising a programming environment having a plurality of programming functionalities. (Mukherjee, col. 5, line 29 – col. 6, line 4: "Display generator 123 operates under the control of application-specific shell 121, and may be implemented in C++, Visual Basic, JAVA, or another high level language."); Figures 3A-3L show a programming

environment supporting certain types of forms that allow a browser to be used for the user interface). Mukherjee also teaches in col. 5, line 46 – col. 6, line 18: after a user has entered information in response to the dynamically changing prompts, forms could be electronically generated and stored for electronic filing. However, Mukherjee does not teach the user interface and controller in communication with a word processing application to control the word processing application to prepare and generate the document. However, HotDocs teaches in lines 22-25 of page 2 controlling a word processing application such as Microsoft Word to prepare and generate a document. Moreover, one of ordinary skill in the art would have recognized that Microsoft Word was a program which many users had on their computers, were comfortable using, and also that the format of a Microsoft Word document would be a format that could easily be shared and disseminated. Therefore, it would have been obvious to one of ordinary skill in the art to have the user interface and controller in communication with a word processing application to control the word processing application to prepare and generate the document.

(B) Mukherjee program is not capable of actually using the information to draft the contents of the document it self.

As to point (B), Mukherjee teaches based on answers/inputs entered by the user during the session, a set of output forms (draft the contents of the document) can be generated

in accordance with the form definition information stored in database 229 (col. 17, line 50 – col. 18, line 7).

(C) The prior art systems are not capable of efficiently preparing a complex document because they do not have the ability to utilize all the functionalities of a full programming environment and database.

As to point (C), Mukherjee teaches in col. 5, line 29 – col. 6, line 4: "Display generator 123 operates under the control of application-specific shell 121, and may be implemented in C++, Visual Basic, JAVA, or another high level language."; Figures 3A-3L show a programming environment supporting certain types of forms that allow a browser to be used for the user interface.. Mukherjee teaches in col. 5, lines 35-40: "In summary, display generator 123 generates graphical user interface components on computer display 130 in response to form definitions retrieved through SQL access component 124 and the firing of expert system rules executed by inference engine 125, which are in turn driven by user inputs received through computer display 130."; see also Mukherjee, col. 2, lines 45-51. In addition, Mukherjee teaches in col. 5, line 46 – col. 6, line 18: after a user has entered information in response to the dynamically changing prompts, forms could be electronically generated and stored for electronic filing.

Art Unit: 2176

(D) The prior art system are not capable of using combinations of logical functions or using such logical functions in a manner such that logical functions are nested within each other.

As to point (D), Mukherjee teaches in col. 10, lines 12-29 and Fig. 3A-3L: showing how repeated or nested data fields can be dynamically generated in response to user selection of a number.

16. Applicant's arguments filed 12/24/2003 have been fully considered but they are not persuasive. Please see the rejection and response to arguments above.

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chau Nguyen whose telephone number is (703) 305-4639. The examiner can normally be reached at 8:00 am – 5:00 pm Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Feild can be reached on (703) 305-9792. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 872-9306. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3230.

Any response to this final action should be mailed to:

**Box A**

Commissioner of Patents and Trademarks  
Washington, D.C. 20131

**Or Faxed to:**

(703) 872-9306, (for **formal communications**; please mark  
“EXPEDITE PROCEDURE”).

**Or:**

(703) 746-7240 (for **informal or draft communications**, please label  
“PROPOSED” or “DRAFT”).

**Or:**

(703) 872-9306 (for **After Final Communications**).

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA., Sixth Floor (Receptionist).

Chau Nguyen  
Patent Examiner  
Art Unit 2176

  
**JOSEPH FEILD**  
**SUPERVISORY PATENT EXAMINER**